Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №4**

**«РАБОТА В ГРАФИЧЕКОМ РЕЖИМЕ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Усатова Ульяна Николаевна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

В отчете должны отображаться:

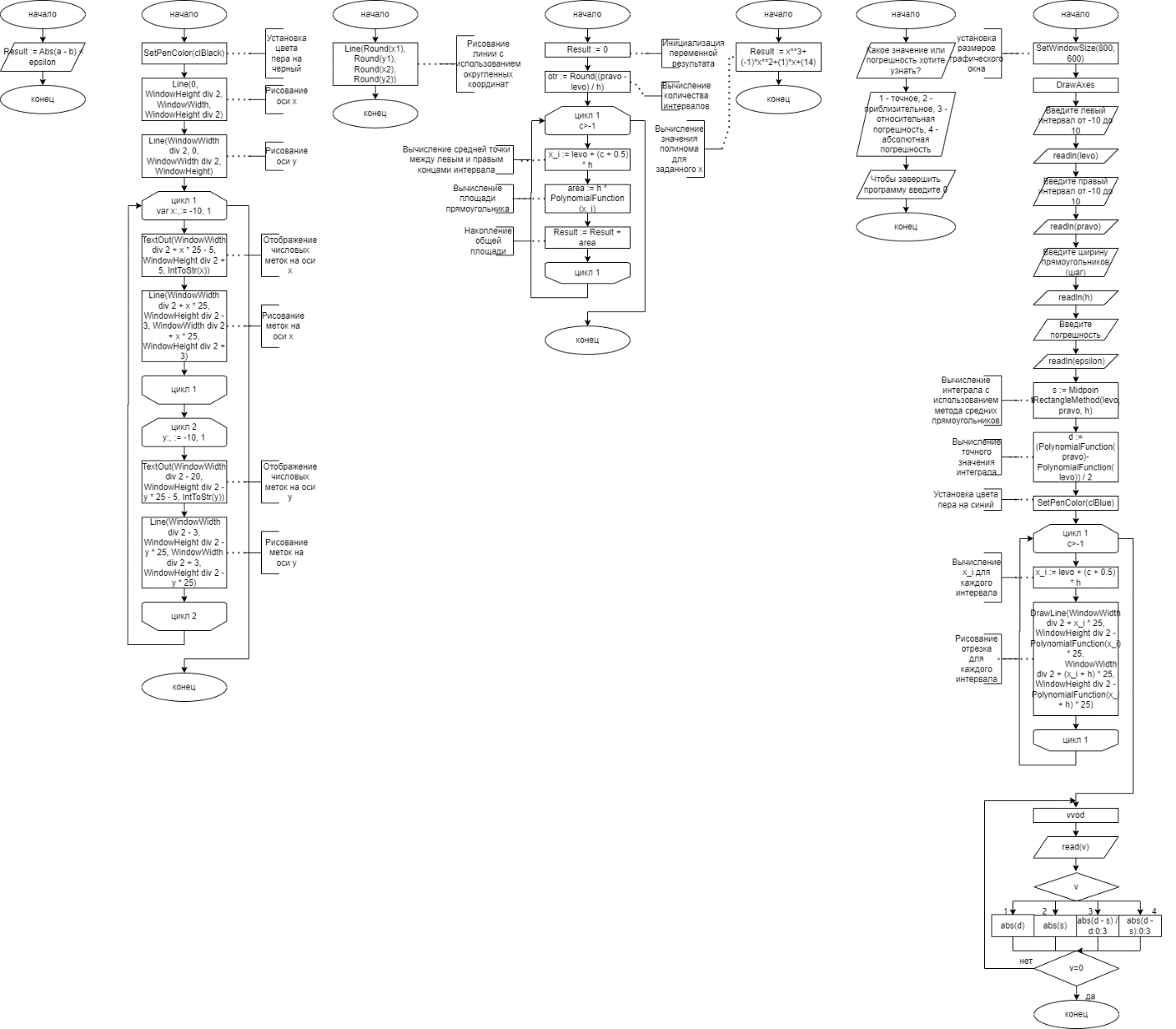
1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод
8. **Цель работы**: освоить принципы работы в графическом режиме, получая базовые навыки взаимодействия с графическими примитивами.
9. **Формулировка задания с вариантом**

Вариант: 20

1. Дополнить программу, реализованную в ходе предыдущей лабораторной работы, режимом визуализации.
2. ﻿﻿﻿Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость.
3. ﻿﻿﻿Реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации о задании
4. ﻿﻿﻿Реализовать не менее двух возможностей из представленных независимое масштабирование по осям, штриховка вычисляемой площади, визуализация численного расчета интеграла.
5. **Описание алгоритма:**

Данный алгоритм состоит из двух процедур. Процедура inf предназначена для выведения информации в модуль clrt. Вторая процедура graph, предназначена для графического отображения графика и выделения площади криволинейной трапеции. Так же задействована процедура left\_method, которая была выполнена ранее. В итоге создано кейс-меню, из которого можно управлять всеми действиями в программе.

1. **Схема алгоритма:**

****

Рисунок

**5.код программы**

**uses**

GraphABC; // Подключение библиотеки GraphABC для работы с графикой

**function** CompareWithEpsilon(a, b, epsilon: real): boolean; // Определение функции сравнения двух вещественных чисел с заданной погрешностью

**begin**

Result := Abs(a - b) < epsilon; // Возвращает true, если абсолютная разница между a и b меньше epsilon

**end**;

**function** PolynomialFunction(x: real): real; // Определение функции для полинома от x

**begin**

Result := x\*\*3+(-1)\*x\*\*2+(1)\*x+(14); // Вычисление значения полинома для заданного x

**end**;

**procedure** DrawLine(x1, y1, x2, y2: real); // Определение процедуры для рисования линии на графическом окне

**begin**

Line(Round(x1), Round(y1), Round(x2), Round(y2)); // Рисование линии с использованием округленных координат

**end**;

**procedure** DrawAxes; // Определение процедуры для рисования осей x и y на графическом окне

**begin**

SetPenColor(clBlack); // Установка цвета пера на черный

Line(0, WindowHeight **div** 2, WindowWidth, WindowHeight **div** 2); // Рисование оси x

Line(WindowWidth **div** 2, 0, WindowWidth **div** 2, WindowHeight); // Рисование оси y

// Рисование меток на оси x

**for var** x: integer := -10 **to** 10 **do**

**begin**

TextOut(WindowWidth **div** 2 + x \* 25 - 5, WindowHeight **div** 2 + 5, IntToStr(x)); // Отображение числовых меток на оси x

Line(WindowWidth **div** 2 + x \* 25, WindowHeight **div** 2 - 3, WindowWidth **div** 2 + x \* 25, WindowHeight **div** 2 + 3); // Рисование меток на оси x

**end**;

// Рисование меток на оси y

**for var** y: integer := -10 **to** 10 **do**

**begin**

TextOut(WindowWidth **div** 2 - 20, WindowHeight **div** 2 - y \* 25 - 5, IntToStr(y)); // Отображение числовых меток на оси y

Line(WindowWidth **div** 2 - 3, WindowHeight **div** 2 - y \* 25, WindowWidth **div** 2 + 3, WindowHeight **div** 2 - y \* 25); // Рисование меток на оси y

**end**;

**end**;

**procedure** vvod; // Определение процедуры для вывода инструкций по вводу

**begin**

writeln('Какое значение или погрешность хотите узнать?'); // Вывод текстовой подсказки на русском

writeln('1 - точное, 2 - приблизительное, 3 - относительная погрешность, 4 - абсолютная погрешность '); // Вывод текстовой подсказки на русском

writeln('Чтобы завершить программу введите 0'); // Вывод текстовой подсказки на русском

**end**;

**function** MidpointRectangleMethod(levo, pravo, h: real): real; // Определение функции для метода средних прямоугольников

**var**

x\_i, area: real; // Объявление переменных для x\_i и площади

c, otr: integer; // Объявление переменных для управления циклом и интервалов

**begin**

Result := 0; // Инициализация переменной результата

otr := Round((pravo - levo) / h); // Вычисление количества интервалов

**for** c := 0 **to** otr - 1 **do**

**begin**

x\_i := levo + (c + 0.5) \* h; // Вычисление средней точки между левым и правым концами интервала

area := h \* PolynomialFunction(x\_i); // Вычисление площади прямоугольника

Result := Result + area; // Накопление общей площади

**end**;

**end**;

**var**

levo, pravo, s, d, h, x\_i: real; // Объявление переменных для границ интервала, результата, шага и x\_i

c, otr, v: integer; // Объявление переменных для управления циклом и пользовательского ввода

epsilon: real; // Объявление переменной для допустимой погрешности

**begin**

SetWindowSize(800, 600); // Установка размеров графического окна

DrawAxes; // Вызов процедуры для рисования осей x и y

writeln('Введите левый интервал от -10 до 10 = '); // Вывод текстовой подсказки на русском

readln(levo); // Считывание пользовательского ввода для левой границы интервала

writeln('Введите правый интервал от -10 до 10 = '); // Вывод текстовой подсказки на русском

readln(pravo); // Считывание пользовательского ввода для правой границы интервала

write('Введите ширину прямоугольников (шаг) = '); // Вывод текстовой подсказки

readln(h); // Считывание пользовательского ввода для шага

write('Введите погрешность = '); // Вывод текстовой подсказки

readln(epsilon); // Считывание пользовательского ввода для допустимой погрешности

s := MidpointRectangleMethod(levo, pravo, h); // Вычисление интеграла с использованием метода средних прямоугольников

d := (PolynomialFunction(pravo) - PolynomialFunction(levo)) / 2; // Вычисление точного значения интеграла

// Рисование кривой функции с использованием метода средних прямоугольников

SetPenColor(clBlue); // Установка цвета пера на синий

**for** c := 0 **to** otr - 1 **do**

**begin**

x\_i := levo + (c + 0.5) \* h; // Вычисление x\_i для каждого интервала

DrawLine(WindowWidth **div** 2 + x\_i \* 25, WindowHeight **div** 2 - PolynomialFunction(x\_i) \* 25,

WindowWidth **div** 2 + (x\_i + h) \* 25, WindowHeight **div** 2 - PolynomialFunction(x\_i + h) \* 25); // Рисование отрезка для каждого интервала

**end**;

**repeat** // Начало цикла для пользовательского ввода

vvod; // Вызов процедуры для отображения инструкций по вводу

read(v); // Считывание пользовательского ввода

**case** v **of** // Начало оператора выбора на основе ввода пользователя

1:

**begin**

writeln('точное значение = ', abs(d):0:3); // Вывод точного значения интеграла

**end**;

2:

**begin**

writeln('приблизительное значение = ', abs(s):0:3); // Вывод приближенного значения интеграла

**end**;

3:

**begin**

writeln('относительная погрешность = ', abs(d - s) / d:0:3); // Вывод относительной погрешности приближенного значения

**end**;

4:

**begin**

writeln('абсолютная погрешность = ', abs(d - s):0:3); // Вывод абсолютной погрешности приближенного значения

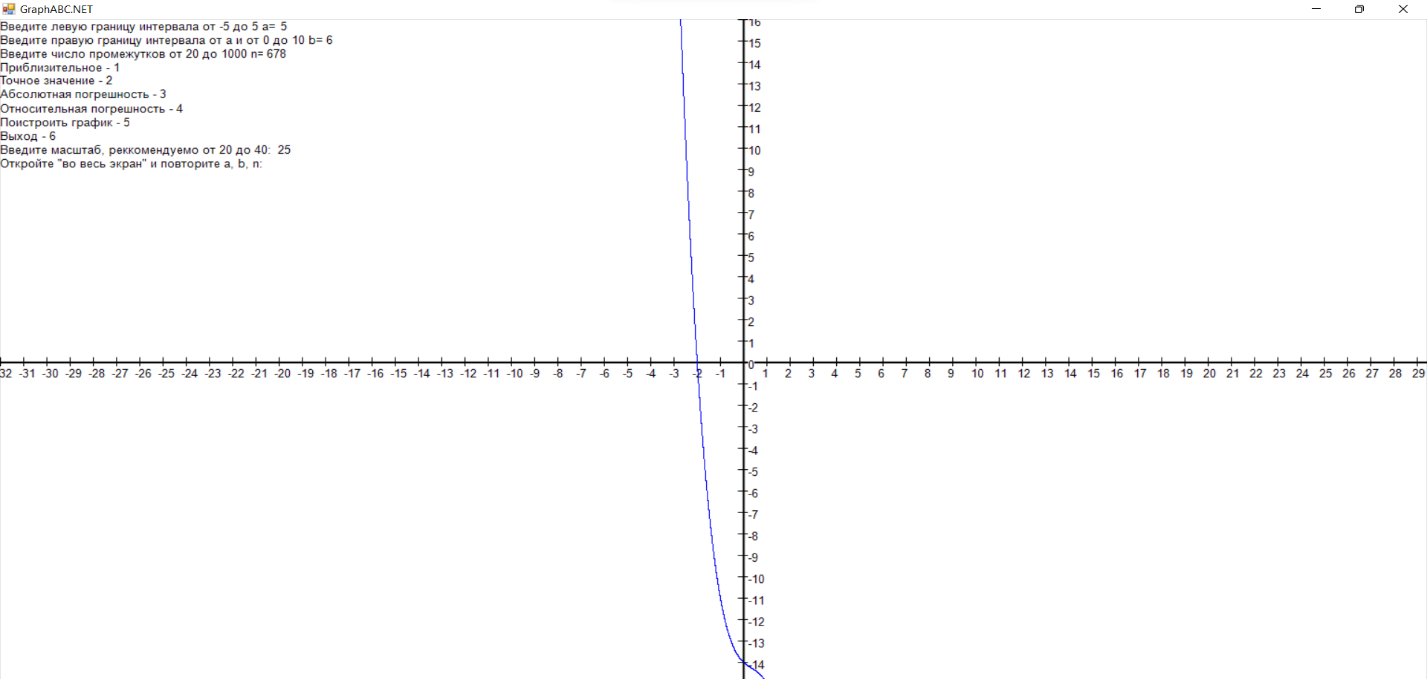
**end**;

**end**;

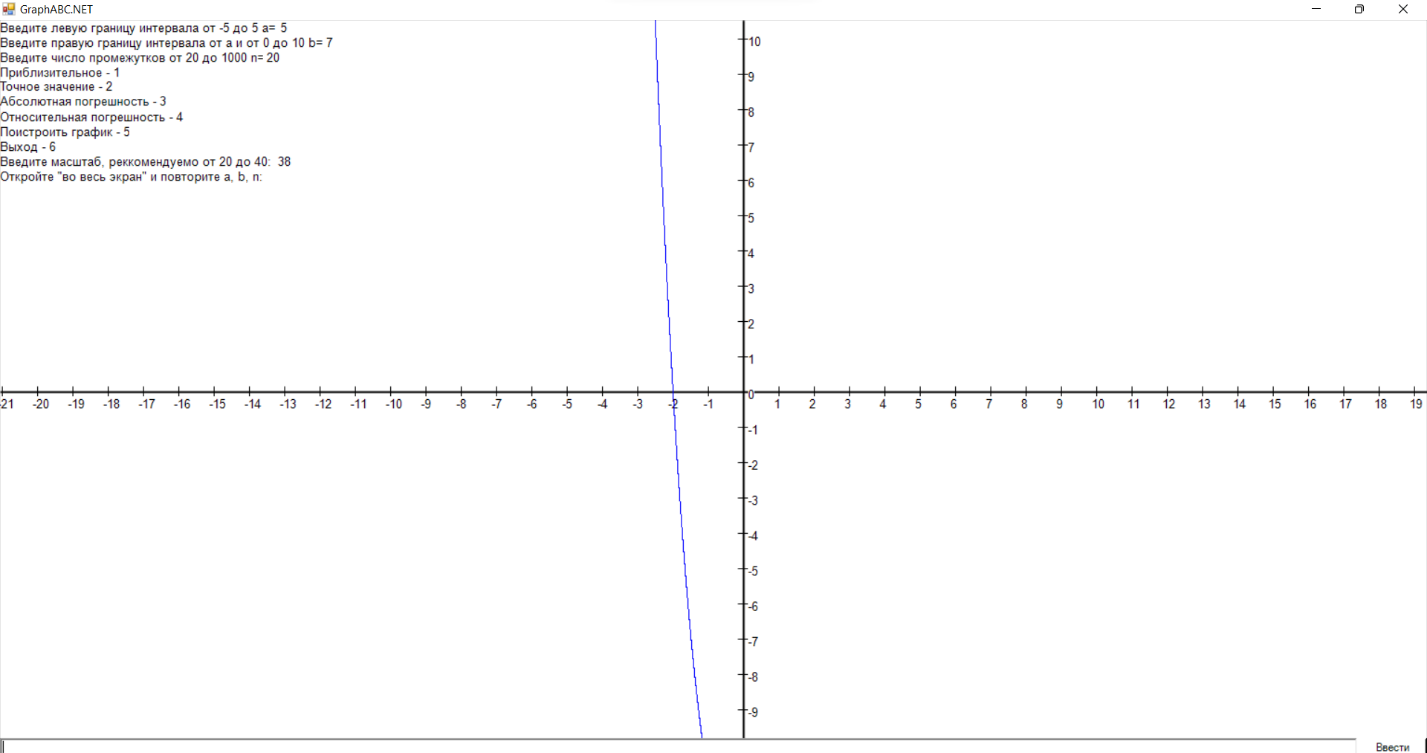
**until** v = 0; // Продолжение цикла до тех пор, пока пользователь не введет 0

**end**.

**6. результат**

****

Рисунок



Рисунок

**7.вывод**

При правильной работе над данной домашней контрольной работой, можно овладеть новыми и полезными знаниями, а результатами программы будет, красивый график с визуализацией функции. В целом – это задание стимулирует получение знаний о продвинутой структуре работы с графикой языка программирования Pascal, о наиболее часто возникающих ошибках с новыми данными. В ходе работы была получена программа, которая визуализирует график функции методом трапеции. Задания были выполнены с помощью графических рисунков на коде Pascal. Был освоен код, который может выводить определенный текст в GraphABC. В результате получены программы, посредством использования графических рисунков, вырисовывается график функции, методом трапеции. После такой работы всегда возникает желание сильнее углубиться в изучаемую тему, к тому же появляется мощная мотивация разбираться в программировании при таких запутанных и интересных домашних контрольных работах.